Representation and Structure in Connectionist Models

Jeffrey L. Elman

August 1989
CRL Technical Report 8903

Center for Research in Language

University of California, San Diego
La Jolla, CA 92093-0108

Representation and Structure in Connectionist Models

Jeffrey L. Elman
Departments of Cognitive Science and Linguistics
University of California, San Diego

ABSTRACT

This paper focuses on the nature of representations in connectionist models. It addresses two issues: (1) Can connectionist models develop representations which possess internal structure and which provide the basis for productive and systematic behavior; and (2) Can representations which are fundamentally context-sensitive support grammatical behavior which appears to be abstract and general? Results from two simulations are reported.. The simulations address problems in the distinction between *type and token*, the representation of *lexical categories*, and the representation of *grammatical structure*. The results suggest that connectionist representations can indeed possess internal structure and enable systematic behavior, and that a mechanism which is sensitive to context is capable of capturing generalizations of varying degrees of abstractness.

INTRODUCTION

Connectionist models appear to provide a new and different framework for understanding cognition. It is therefore natural to wonder how these models might differ from traditional theories, and what their advantages or disadvantages might be. Recent discussion has focussed on a number of topics, including the treatment of regular and productive behavior (rules vs. analogy), the form of knowledge (explicit vs. implicit), the ontogeny of knowledge (innate vs. acquired), and the nature of connectionist representations.

This latter issue is particularly important because one of the critical ways in which cognitive theories may differ is in the representational apparatus they make available. Our current understanding of connectionist representations is at best partial, and there is considerable diversity of opinion among those who are actively exploring the topic (cf. Dolan & Dyer, 1987; Dolan & Smolensky, 1988; Feldman & Ballard, 1982; Hanson & Burr, 1987; McMillan & Smolensky, 1988; Hinton, 1988; Hinton, McClelland, & Rumelhart (1986); McClelland, St. John, & Taraban (1989); Pollack, 1988; Ramsey, 1989; Rumelhart, Hinton, & Williams, 1986; Shastri & Ajjanagadde, 1989; Smolensky, 1987a, 1987b, 1987c, 1988; Touretzky & Hinton, 1985; Touretzky, 1986, 1989; van Gelder, in press).

In this paper I would like to focus on

some of the specific questions raised by Fodor & Pylyshyn (1988). Fodor & Pylyshyn express concern that whereas Classical theories (e.g., the Language of Thought, Fodor, 1976) are committed to complex mental representations which reflect combinatorial structure, connectionist representations seem to be atomic, and therefore (given the limited and fixed resources available to them) finite in number. And this appears to be at odds with what we believe to be necessary for human cognition in general, and human language in particular.

I believe that Fodor and Pylyshyn are right in stressing the need for representations which support complex and systematic patterning, which reflect both the combinatorics and compositionality of thought, and which enable an open-ended productions. What their analysis does not make self-evident is that these desiderata can only be achieved by the so-called Classical theories, or by connectionist models which implement those theories. Fodor & Pylyshyn present a regrettably simplistic picture of current linguistic theory. What they call the Classical theory actually encompasses a heterogeneous set of theories, not all of which are obviously compatible with the Language of Thought. Furthermore, there have in recent years been well-articulated linguistic theories which do not share the basic premises of the Language of Thought (e.g., Chafe, 1970; Fauconnier, 1985; Fillmore, 1982; Givon, 1984; Hopper & Thompson, 1980; Kuno, 1987; Lakoff, 1987; Langacker, 1987). Thus the two alternatives presented by Fodor and Pylyshyn (that connectionism must either implement the Language of Thought or fail as a cognitive model) are unnecessarily bleak and do not exhaust the range of possibilities.

Still, it is possible to phrase the questions posed by Fodor & Pylyshyn in a more general way which might be profitably pursued: What is the nature of connectionist representations? Are they necessarily atomistic or can they possess internal structure? Can that structure be used to account for behavior which reflects both general and ideosyncratic patterning? Can connectionist representations with finite resources provide an account for apparently open-ended productive behavior? How might connectionist representations differ from those in the Language of Thought? One strength of connectionist models that is often emphasized is their sensitivity to context and ability to exhibit graded responses to subtle differences in stimuli (e.g., McClelland, St John, & Taraban, 1989). But sometimes language behavior seems to be characterized by abstract patterns which are less sensitive to context. So another question is whether models which are fundamentally context-sensitive are also able to arrive at generalizations which are highly abstract.

In this paper I present results from two sets of simulations. These simulations were designed to probe the above issues, with the goal of providing some insight into the representational capacity of connectionist models. The paper is organized in two sections. The first section reports empirical results. Two connectionist networks were taught tasks in which an abstract structure underlay the stimuli and task. The intent was to create problems which would encourage the development of internal representations which reflected that abstract structure. Both the performance of the networks as well as the analysis of their solutions illustrates the development of internal representations which are richly structured. These results are discussed at greater length in the second section, and related to the broader question of the usefulness of the connectionist framework for modeling cognitive phenomena, and possible differences from the Classical approach.

Part I: SIMULATIONS

Language is structured in a number of ways. One important kind of structure has to do with the structure of the categories of language elements (e.g., words). The first simulation addressed the question of whether a connectionist model can induce the lexical category structure underlying a set of stimuli. A second way in which language is structured has to do with the possible ways in which strings can be combined (e.g., the grammatical structure). The second simulation addresses that issue.

LEXICAL CATEGORY STRUCTURE

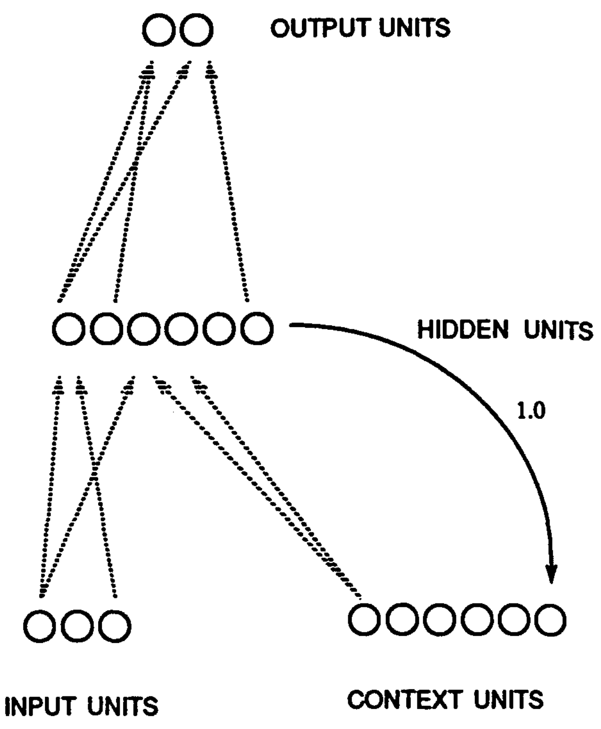
Words may be categorized with respect to many factors. These include traditional notions such as *noun*, *verb*, etc.; the argument structure they are associated with; and their semantic features. One of the consequences of lexical category structure is word order. Not all classes of words may appear in any position. Furthermore, certain classes of words, e.g., transitive verbs, tend to cooccur with other words (as we shall see in the next simulation, these cooccurrence facts can be quite complex).

The goal of the first simulation was to see if a network could learn the lexical category structure which was implicit in a language corpus. The overt form of the language items was arbitrary, in the sense that the form of the lexical items contained no information about their lexical category. However, the behavior of the lexical item—defined in terms of cooccurrence restrictions—reflected their membership in implicit classes and subclasses. The question was whether or not the network could induce these classes.

Network Architecture

Time is an important element in language, and so the question of how to represent serially ordered inputs is crucial. Various proposal have been advanced (for reviews, see Elman, in press; Mozer, 1988. The approach taken here involves treating the network as a simple dynamical system in which previous states are made available as an additional input (Jordan, 1986). In Jordan's work the prior state was derived from the output units on the previous time cycle. In the work here, the prior state comes from the hidden unit patterns on the previous cycle. Because the hidden units are not taught to assume specific values, this means that they can develop representations, in the course of learning a task, which encode the temporal structure of the task. In other words, the hidden units learn to be-

Figure 1.



come a kind of memory which is very task-specific.

The type of network used in the first simulation is shown in Figure 1. This network is basically a 3-layer network with the customary feed-forward connections from **input units** to **hidden units**, and from hidden units to **output units**. There are an additional set of units, called **context units**, which provide for limited recurrence (and so this may be called a **simple recurrent network**). These context units are activated on a one-for-one basis by the hidden units, with a fixed weight of 1.0.

The result is that at each time cycle the hidden unit activations are copied into the context units; on the next time cycle, the context combines with the new input to activate the hidden units. The hidden units therefore take on the job of mapping new inputs and prior states to the output. Because they themselves constitute the prior state, they must develop representations which facilitate this input/output mapping. The simple recurrent network has been studied in a number of tasks (Elman, in press; Hare, Corina, & Cottrell, 1988; Servan-Schreiber, Cleeremans, & McClelland, 1988). In this first simulation, there were 31 input units, 150 hidden and context units, and 31 output units.

Stimuli and Task

A lexicon of 29 nouns and verbs was chosen. Words were represented as 31-bit binary vectors (two extra bits were reserved for another purpose). Each words was randomly assigned a unique vector in which only one bit was turned on. A sentence-generating program was then used to create a corpus of 10,000 2- and 3-word sentences. The sentences reflected certain properties of the words. For example, only animate nouns occurred as the subject of the verb **eat**, and this verb was only followed by edible substances. Finally, the words in successive sentences were concatenated, so that a stream of 27,354 vectors was created This formed the input set.

The task was simply for the network to take successive words from the input stream and to predict the subsequent word (by producing it on the output layer). After each word was input, the output was compared with the actual next word, and the backpropagation of error learning algorithm (Rumelhart, Hinton, & Williams, 1986) was used to adjust the network weights. Words were presented in order, with no breaks between sentences. The network was trained on 6 passes through the corpus.

The prediction task was chosen for several reasons. First, it makes minimal assumptions about special knowledge required for training. The teacher function is simple and the information provided available in the world at the next moment in time. Thus, there are no *a priori* theoretical commitments which might bias the outcome. Second, although the task is simple and should not be taken as a model of comprehension, it does seem to be the case that much of what listeners do involves anticipation of future input (Grosjean, 1980; Marslen-Wilson & Tyler, 1980; Salasoo & Pisoni, 1985).

Results

Because the sequence is non-deterministic, short of memorizing the sequence, the network cannot succeed in exact predictions. That is, the underlying grammar and lexical category structure provides a set of constraints on the form of sentences, but the sentences themselves involve a high degree of optionality. Thus, measuring the performance of the network in this simulation is not straightforward. Root mean squared error at the conclusion of training had dropped to 0.88. However, this result is not impressive. When output vectors are sparse, as those used in this simulation were (only 1 out of 31 output bits was to be turned on), the network quickly learns to reduce error dramatically by turning all the output units off. This drops error from the initial random value of ~15.5 to 1.0, which is close to the final rmse value of 0.88.

Although the prediction task is nondeterministic, it is also true that word order is not random or unconstrained. For any given sequence of words there are a limited number of possible successors. Under these circumstances, it would seem more appropriate to ask whether or not the network has learned what the class of valid successors is, at each point in time. We therefore might expect that the network should learn to activate the output nodes to some value proportional to the probability of occurrence of each word in that context.

Therefore, rather than evaluating final network performance using the rms error calculated by comparing the network's output with the actual next word, we can compare the output with the probability of occurrence of possible successors. These values can be derived empirically from the training data base (for details see Elman, in press); such calculation yields a "likelihood output vector" which is appropriate for each input, and which reflects the context-dependent expectations given the training base (where context is defined as extending from the beginning of the sentence to the input). Note that it is appropriate to use these likelihood vectors only for the evaluation phase. Training must be done on the actual successor words because the point is to force the network to learn the context-dependent probabilities for itself.

Evaluated in this manner, the error on the training set is 0.053 (sd: 0.100). The cosine of the angle between output vectors and likelihood vectors provides another measure of performance (which normalizes for length differences in the vectors); the mean cosine is 0.916 (sd: 0.123), indicating that the two vectors on average have very similar shapes. Objectively, the performance appears to be quite good.

Lexical categories

The question to be asked now is how this performance has been achieved. One way to answer this is to see what sorts of internal representations the network develops in order to carry out the prediction task. This is particularly relevant, given the focus of the current paper. The internal representations are instantiated as activation patterns across the hidden units which are evoked in response to each word in its context. These patterns were saved during a testing phase during which no learning took place. For each of the 29 unique words a mean vector was then computed which averaged across all occurrences of the word in various contexts. These mean vectors were then subjected to hierarchical clustering analysis. Figure 2 shows the tree constructed from the hidden unit patterns for the 29 lexical items.

The tree in Figure 2 shows the similarity structure of the internal representations of the 29 lexical items. The form of each item is randomly assigned (and orthogonal to all

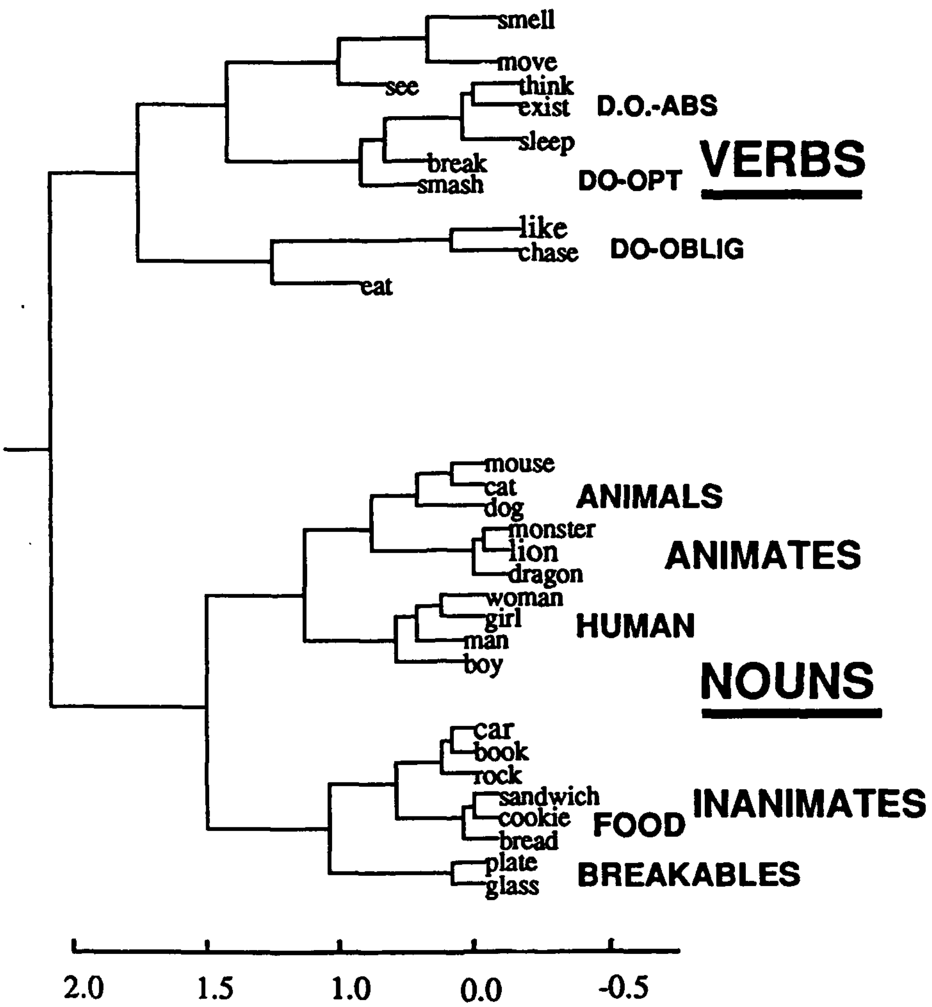


Figure 2

other items), and so the basis for the similarity in the internal representations is the way in which these words "behave" with regard to the task.

The network has discovered that there are several major categories of words. One large category corresponds to *verbs;* another category corresponds to *nouns.* The verb category is broken down into groups which require a direct object; which are intransitive; and for which a direct object is optional. The noun category is divided into major groups for *animates* and *inanimates.* Animates are divided into *human* and *non-human;* the *non-humans* are sub-divided into *large animals* and *small animals.* Inanimates the divided into *breakables, edibles,* and miscellaneous.

This category structure reflects facts about the possible sequential ordering of the inputs. The network is not able to predict the precise order of specific words, but it recognizes that (in this corpus) there is a class of inputs (*viz.*, verbs) which typically follow other inputs (*viz.*, nouns). This knowledge of class behavior is quite detailed; from the fact that there is a class of items which always precedes **chase**, **break**, and **smash**, it infers a category of large animals (or possibly, aggressors).

Several points should be emphasized. First, the category structure appears to be hierarchical. **Dragons** are large animals, but also members of the class [-human, +animate] nouns. The hierarchical interpretation is achieved through the way in which the spatial relations of the representations are organized. Representations which are near one another in representational space form classes, and higher-level categories correspond to larger and more general regions of this space.

Second, it is also the case that the hierarchicality and category boundaries are "soft". This does not prevent categories from being qualitatively distinct by being far from each other in space with no overlap. But there may also be entities which share properties of otherwise distinct categories, so that in some cases category membership may be marginal or ambiguous.

Finally, the content of the categories is not known to the network. The network has no information available which would ground the structural information in the real world. This is both a plus and a minus. Obviously, a full account of language processing needs to provide such grounding. On the other hand, it is interesting that the evidence for category structure can be inferred so readily on the basis of language-internal evidence alone.

Type-token distinctions

The tree shown in Figure 2 was constructed from activation patterns averaged across context. It is also possible to cluster activation patterns evoked in response to words in the various contexts in which they occur. When the context-sensitive hidden units patterns are clustered, it is found that the large-scale structure of the tree is identical to that shown in Figure 2. However, each terminal leaf is now replaced with further arborization for all occurrences of the word (there are no instances of lexical items appearing on inappropriate branches).

This finding bears on the type/token problem in an important way. In this simulation, the context makes up an important part of the internal representation of a word. Indeed, it is somewhat misleading to speak of the hidden unit representations as word representations in the conventional sense, *since these* patterns also reflect the prior context. As a result, it is literally the case that every occurrence of a lexical item has a separate internal representation. We cannot point to a canonical representation for **John**; instead there are representations for **John$_1$**, **John$_2$**, ... **John$_n$**. These are the tokens of **John**, and the fact that they are different is the way the system marks what may be subtle but important meaning differences associated with the specific token. The fact that these are all tokens of the same type is not lost, however. These tokens have representations which are extremely close in space — closer to each other by far than to any other entity. Even more interesting is that the spatial organization within the token space is not random but reflects differences in context which are also found among tokens of other items. The tokens of **boy** which occur in subject position tend to cluster together, as distinct from tokens of **boy** which occur in object position. This distinction is marked in the same way for tokens of other nouns. Thus, the network has learned not only about types and tokens, and categories and category members; it also has learned a grammatical role distinction which cuts across lexical items.

This simulation has involved a task in which the category structure of inputs was an important determinant of their behavior. The category structure was apparent in their behavior only; their external form provided no useful information. We have seen that the network makes use of spatial organization in order to capture this category structure.

We turn next to a problem in which the lexical category structure provides only one part of the solution, and in which the network must learn abstract grammatical structure.

Representation of grammatical structure

In the previous simulation there was little interesting structure of the sort that related words to one another. Most of the relevant information regarding sequential behavior was encoded in terms of invariant properties of items. Although lexical information plays an important role in language, it actually accounts for only a small range of facts. Words are processed in the contexts of other words; they inherit properties from the specific grammatical structure in which they occur. This structure can be quite complex, and it is not clear that the kind of category structure supported by the spatial distribution of representations is sufficient to capture the structure which belongs, not to individual words, but to particular configurations of words.

As we consider this issue, we also note that till now we have neglected an important dimension along which structure may be manifest, *time*. The clustering technique used in the previous simulation informs us of the similarity relations along spatial dimensions. The technique tells us nothing about the patterns of movement through space. This is unfortunate, since the networks we are using are dynamical systems whose states change over time. Clustering groups states according to the metric of Euclidean distance but in so doing discards the information about whatever temporal relations may hold between states. This information is clearly relevant if we are concerned about grammatical structure. Consider the sentences

(1a) The man saw the **car**.

(1b) The man who saw the **car** called the cops.

On the basis of the results of the previous simulation, we would expect that the representations for the word **car** in these two sentences would be extremely similar. Not only are they the same lexical type, but they both appear in clause-final position as the object of the same verb.

But we might also wish to have their representations capture an important structural difference between them. **Car** in sentence (1a) occurs at the end of the sentence; it brings us to a state from which we should move into another class of states that are associated with the onsets of new sentences. In sentence (1b), **car** is also at the end of a clause, but occurs in a matrix sentence which has not yet been completed. There are grammatical obligations which remain unfulfilled. We would like the state that is associated with **car** in this context to lead us to the class of states which might conclude the main clause.

The issue of how to understand the

temporal structure of state trajectories will thus figure importantly in our attempts to understand the representation of grammatical structure.

Stimuli and Task

The stimuli in this simulation were based on a lexicon of 23 items. These included 8 nouns, 12 verbs, the relative pronoun **who**, and an end-of-sentence indicator, "." . Each item was represented by a randomly assigned 26-bit vector in which a single bit was set to 1 (3 bits were reserved for another purpose). A phrase structure grammar, shown in Table 1, was used to generate sentences. The resulting sentences possessed certain important properties. These include the following.

(a) Agreement

Subject nouns agree with their verbs. Thus, tor example, (2a) is grammatical but not (2b) (the training corpus consisted of positive examples only; thus the starred examples below did not occur).

(2a) John feeds dogs.

(2b) *Boys sees Mary.

Words are not marked for number (singular/plural), form class (verb/noun, etc.), or grammatical role (subject/object, etc.). The network must learn first that there are items which function as what we would call nouns, verbs, etc.; then it must learn which items are examples of singular and plural; and then it must learn which nouns are

S → NP VP "."
NP → PropN | N | N RC
VP → V (NP)
RC → *who* **NP VP |** *who* **VP (NP)**
N → *boy* | *girl* | *cat* | *dog* | *boys* | *girls* | *cats* | *dogs*
PropN → *John* | *Mary*
V → *chase* | *feed* | *see* | *hear* | *walk* | *live* | *chases* | *feeds* | *sees* | *hears* | *walks* | *lives*

Additional restrictions:

- **number agreement between N & V within clause, and (where appropriate) between head N & subordinate V**

- **verb arguments:**

  hit, feed → **require a direct object**
  see, hear → **optional allow a direct object**
  walk, live → **preclude a direct object**
  (observed also for head/verb relations in relative clauses)

Table 1

subjects and which are objects (since agreement only holds between subject nouns and their verbs).

(b) Verb argument structure

Verbs fall into three classes: those that require direct objects, those that permit an optional direct object, and those that preclude direct objects. As a result, sentences (3a-d) are grammatical, whereas sentences (3e, 3f) are ungrammatical.

(3a) Girls feed dogs. (*D.o. required*)

(3b) Girls see boys. (*D.o. optional*)

(3c) Girls see. (*D.o. optional*)

(3d) Girls live. (*D.o. precluded*)

(3e) *Girls feed.

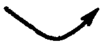
(3f) *Girls live dogs.

Again, the type of verb is not overtly marked in the input, and so the class membership needs to be inferred at the same time as the cooccurrence facts are learned.

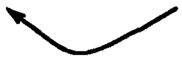
(c) Interactions with relative clauses

Both the agreement and the verb argument facts are complicated in relative clauses. While direct objects normally follow the verb in simple sentences, some relative clauses have ihe direct object as the head of the clause, in which case the network must learn to recognize that the direct object has already been filled (even though it occurs before the verb). T:us, the normal pattern in simple sentences (3a-d) appears also in (4a), but contrasts with (4b),

(4a) Dog who chases cat sees girl.



(4b) Dog who cat chases sees girl.

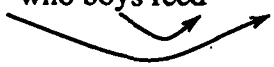


Sentence (4c), which seems to conform to the pattern established in (3), is ungrammatical.

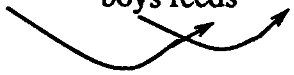
(4c) *Dog who cat chases dog sees girl.

Similar complications arise for the agreements facts. In simple sentences agreement involves *N1* - *V1*. In complex sentences, such as (5a), that regularity is violated, and any straightforward attempt to generalize it to sentences with multiple clauses would lead to the ungrammatical (5b).

(5a) Dog who boys feed sees girl.



(5b) *Dog who boys feeds see girl.



(d) Recursion

The grammar permits recursion through the presence of relative clause (which expand to noun phrases which may introduce yet other relative clauses, etc.). This leads to sentences such as (6) in which the grammatical phenomena noted in (*a-c*) may be extended over a considerable distance.

(6) Boys who girls who dogs chase see hear.

(e) Viable sentences

One of the literals inserted by the grammar is

".", which occurs at the end of sentences. This end-of-sentence marker can of course potentially occur anywhere in a string where a sentence is viable (in the sense that it is grammatically well-formed and may at that point be terminated). Thus in sentence (7), the arrows indicate positions where a "." might legally occur.

(7) Boys see dogs who see girls who hear.
↑ ↑ ↑
* * *

The data in (4-7) are examples of the sorts of phenomena which linguists argue cannot be accounted for without abstract representations; it is these representations rather than the surface strings on which the correct grammatical generalizations are made.

A network of the form shown in Figure 3 was trained on the prediction task (layers are shown as rectangles; numbers indicate the number of nodes in each layer).

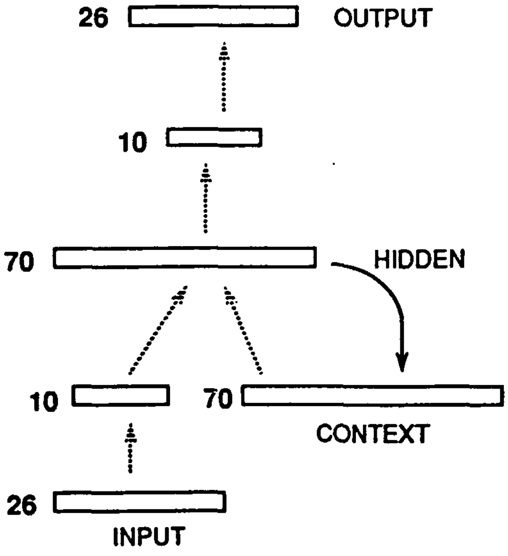


Figure 3

The training data were generated from the phrase structure grammar given in Table 1. At any given point during training, the training set consisted of 10,000 sentences which were presented to the network 5 times. (As before, sentences were concatenated so that the input stream proceeded smoothly without breaks between sentences.) However, the composition of these sentences varied over time. The following training regimen was used in order to provide for incremental training. The network was trained on 5 passes through each of the following 4 corpora.

Phase 1: The first training set consisted exclusively of simple sentences. This was accomplished by eliminating all relative clauses. The result was a corpus of 34,605 words forming 10,000 sentences (each sentence includes the terminal ".").

Phase 2: The network was then exposed to a second corpus of 10,000 sentences *which consisted of 25%* complex sentences and 75% simple sentences (complex sentences were obtained by permitting relative clauses). Mean sentence length was 3.92 (minimum 3 words, maximum 13 words).

Phase 3: The third corpus increased the percentage of complex sentences to 50%, with mean sentence length of 4.38 (minimum: 3 words, maximum: 13 words).

Phase 4: The fourth consisted of 10,000 sentences, 75% complex, 25% simple. Mean sentence length was 6.02 (minimum: 3 words, maximum: 16 words).

This staged learning strategy was developed in response to results of earlier pilot work. In this work, it was found that the network was unable to learn the task when given the full range of complex data from the beginning of training. However, when the network was permitted to focus on the simpler data first, it was able to learn the task quickly and then move on successfully to more complex patterns. The important aspect to this was that the earlier training constrained later learning in a useful way; the early training forced the network to focus on canonical versions of the problems which apparently created a good basis for then solving the more difficult forms of the same problems.

Results

At the conclusion of the fourth phase of training, the weights were frozen at their final values and network performance was tested on a novel set of data, generated in the same way as the last training corpus. The technique described in the previous simulation was used; context-dependent likelihood vectors were generated for each word in every sentences. These vectors represented the empirically derived probabilities of occurrence for all possible predictions, given the sentence context up to that point. The rms error of network predictions, compared against the likelihood vectors, was 0.177 (sd: 0.463); the mean cosine of the angle between the vectors was 0.852 (sd: 0.259). Although this performance is not as good as in the previous simulation, it is still quite good. And the task is obviously much more difficult.

These gross measures of performance however do not tell us how well the network has done in each of the specific problem areas posed by the task. Let us look at each area in turn.

(a) Agreement in simple sentences

Agreement in simple sentences is shown in Figures 4a and 4b.

The network's predictions following the word **boy** are that either a singular verb will follow (words in all three singular verb categories are activated, since it has no basis for predicting the type of verb), or else that the next word may be the relative pronoun **who**. Conversely, when the input is the word **boys**, the expectation is that a verb in the plural will follow, or else the relative pronoun. Similar expectations hold for the other nouns in the lexicon.

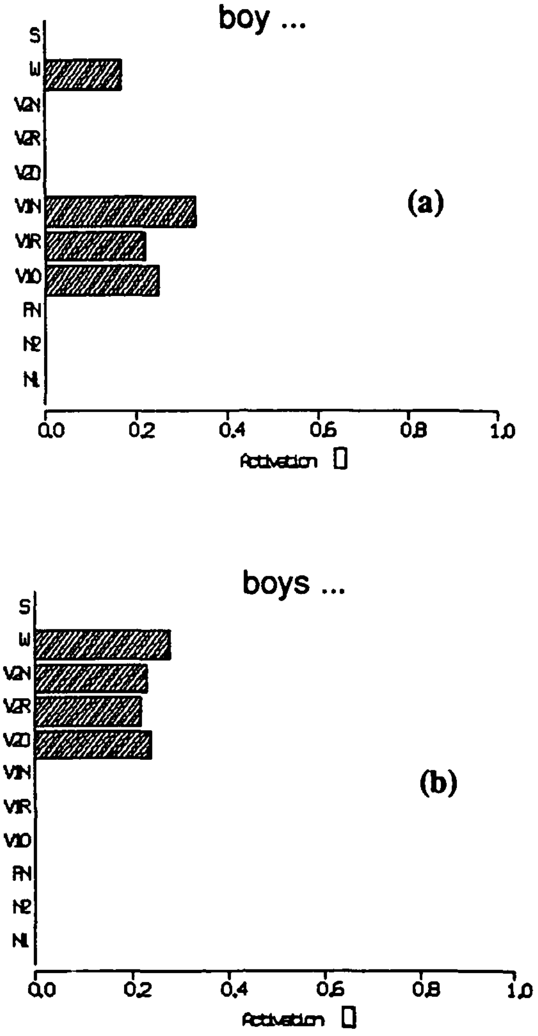
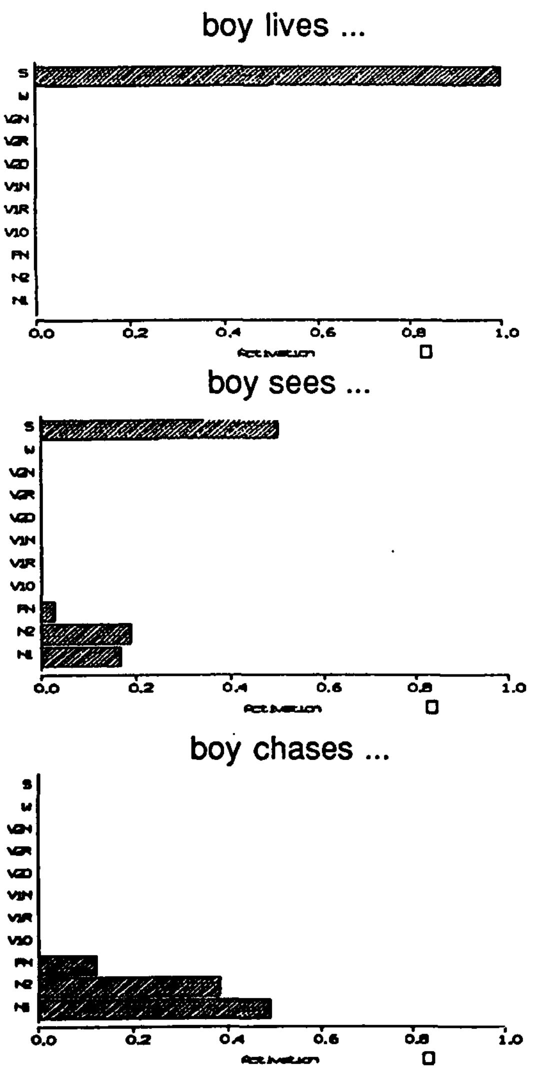


Figure 4
(a) Graph of network predictions following presentation of the word boy. Predictions are shown as activations for words grouped by category. S stands for end-of-sentence ("."); W stands for who; N and V represent nouns and verbs; 1 and 2 indicates singular or plural; and type of verb is indicated by N, R, O (direct object not possible, required, or optional). (b) Graph of network predictions following presentation of the word boys.

(b) Verb argument structure in simple sentences

Figure 5 shows network predictions following an initial noun and then a verb from each of the three different verb types.

When the verb is **lives,** the network's expectation is that the following item will be "." (which is in fact the only successor permitted by the grammar in this context). The verb **sees,** on the other hand, may either be followed by a ".", or optionally by a direct object (which may be a singular or plural noun, or proper noun). Finally, the verb **chases** requires a direct object, and the network learns to expect a noun following this and other verbs in the same class.



boy lives ...

boy sees ...

boy chases ...

Figure 5

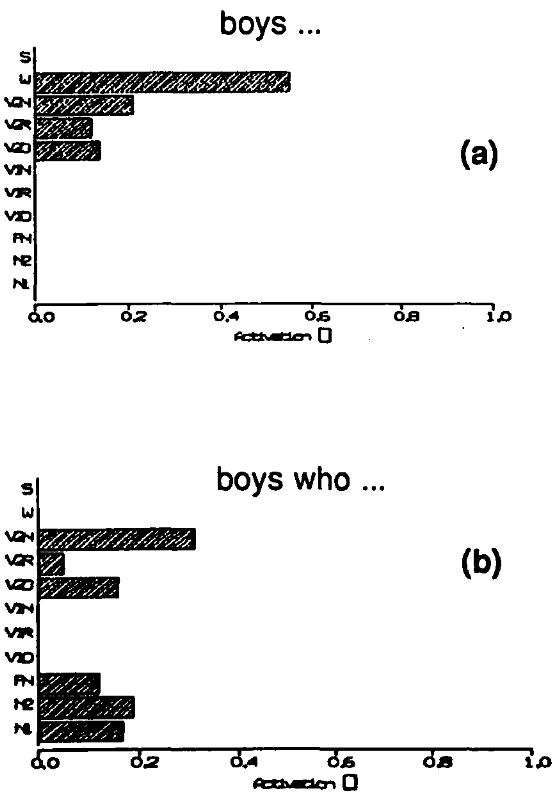
(c) Interactions with relative clauses

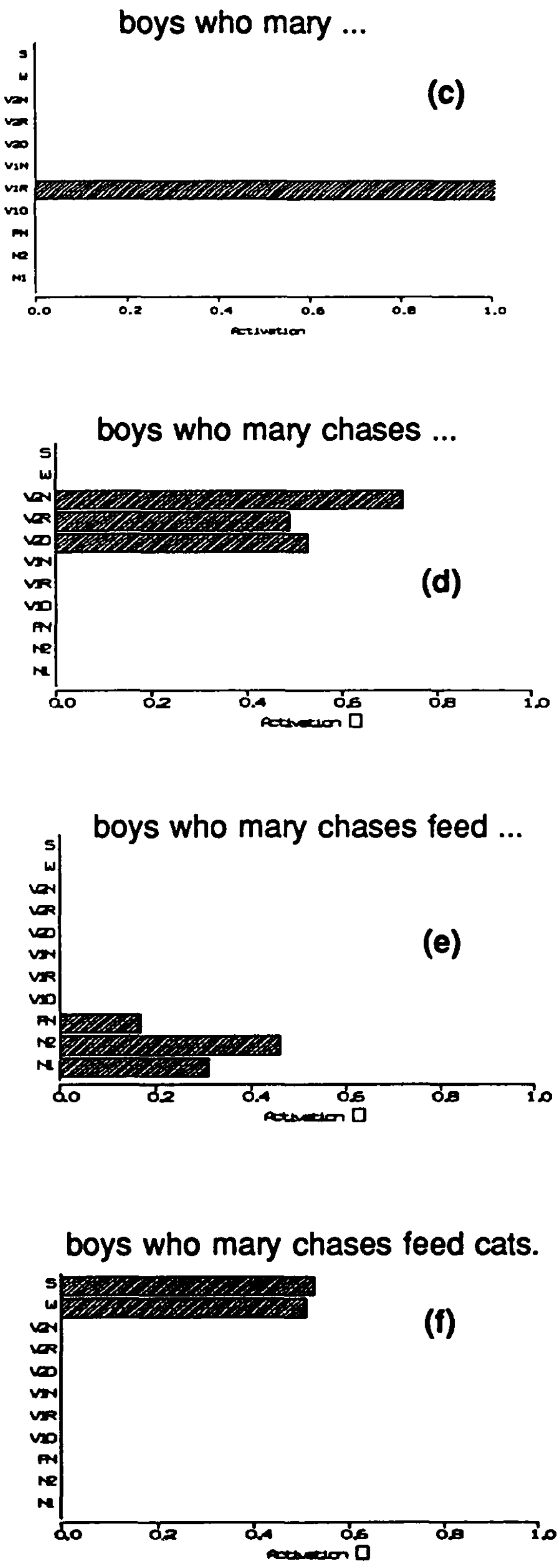
The examples so far have all involved simple sentences. The agreement and verb argument facts are more complicated in complex sentences. Figure 6 shows the network predictions for each word in the sentence **boys who mary chases feed cats.** If the network were generalizing the pattern for agreement found in the simple sentences, we might expect the network to predict a singular verb following **...mary chases...** (insofar as it predicts a verb in this position at all; conversely, it might be confused by the pattern *N1 N2 V1*). But in fact, the prediction (6d) is correctly that the next verb should be in the singular in order to agree with the first noun. In so doing, it has found some mechanism for representing the long-distance dependency between the main clause noun and main clause verb, despite the presence of an intervening noun and verb (with their own agreement relations) in the relative clause.

Note that this sentence also illustrates the sensitivity to an interaction between verb argument structure and relative clause structure. The verb **chases** takes an obligatory direct object. In simple sentences the direct object follows the verb immediately; this is also true in many complex sentences (e.g., **boys who chase mary feed cats**). In the sentence displayed, however, the direct object (**boys**) is the head of the relative clause and appears before the verb. This requires that the network learn (a) there are items which function as nouns, verbs, etc..; (b) which items fall into which classes; (c) there are subclasses of verbs which have different cooccurrence relations with nouns, corresponding to verb-direct object restrictions; (d) which verbs fall into which classes; and (e) when to expect that the direct object will follow the verb, and when to know that it has already appeared. The network appears to have learned this, because in panel (d) we see that it expects that **chases** will be followed by a verb (the main clause verb, in this case) rather than a noun.

An even subtler point is demonstrated in (6c). The appearance of **boys** followed by a relative clause containing a different subject (**who Mary...**) primes the network to expect that the verb which fol-

Figure 6





lows must be of the class that requires a direct object, precisely because a direct object filler has already appeared. In other words, the network correctly responds to the presence of a filler (**boys**) not only by knowing where to expect a gap (following **chases**); it also learns that when this filler corresponds to the object position in the relative clause, a verb is required which has the appropriate argument structure.

Network analysis

The natural question to ask at this point is how the network has learned to accomplish the task. It was initially assumed that success on this task would constitute *prima facie* evidence for the existence of internal representations which possessed abstract structure. That is, it seemed reasonable to believe that in order to handle agreement and argument structure facts in the presence of relative clauses, the network would be required to develop representations which reflected constituent structure, argument structure, grammatical category, grammatical relations, and number.

Having achieved success on the task, we now would like to test this assumption. In the previous simulation, hierarchical clustering was used to reveal the use of spatial organization at the hidden unit level for categorization purposes. However, the clustering technique makes it difficult to see patterns which exist over time. Some states may have significance not simply in terms of their similarity to other states, but with regard to the ways in which they constrain movement into subsequent state space (recall the examples in (1)). Because clustering ignores the temporal information, it hides this information. What would be more useful would be to look at the trajectories through state space over time which correspond to the internal representations evoked at the hidden unit layer as a network processes a given sentence.

Phase-state portraits of this sort are commonly limited to displaying not more than a few state variables at once, simply because movement in more than three dimensions is difficult to graph. The hidden unit activation patterns in the current simulation take place over 70 variables. These patterns are distributed, in the sense that none of the hidden units alone provides useful information; the information instead lies along hyperplanes which cut across multiple units.

However, it is possible to identify these hyperplanes using principle component analysis. This involved passing thing training set through the trained network (with weights frozen) and saving the hidden unit pattern for produced in response to each new input. The covariance matrix of the set of hidden unit vectors is calculated, and then the eigenvectors for the covariance matrix are found. The eigenvectors are ordered by the magnitude of their eigenvalues, and are used as the new basis for describing the original hidden unit vectors. This new set of dimensions has the effect of giving a somewhat more localized description to the hidden unit patterns, because the new dimensions now correspond to the location of meaningful activity (defined in terms of variance) in the hyperspace. Furthermore, since the dimensions are ordered in terms of variance accounted for, we can now look at phase state portraits of selected dimensions, starting with those with largest eigenvalues.

Agreement

The sentences in (8) were presented to the network, and the hidden unit patterns captured after each word was processed in sequence.

(8a) boys hear boys .

(8b) boy hears boys .

(8c) boy who boys chase chases boy .

(8d) boys who boys chase chase boy .

(These sentences were chosen to minimize differences due to lexical content and to make it possible to focus on differences to grammatical structure. (8a) and (8b) were contained in the training data; (8c) and (8d) were novel and had never been presented to the network during learning.)

By examining the trajectories through state space along various dimensions, it was apparent that the second principle component played an important role in marking number of the main clause subject. Figure 7 shows the trajectories for (8a) and (8b); the trajectories are overlaid so that the differences are more readily seen. The paths are similar and diverge only during the first word, indicating the difference in the number of the initial noun. The difference is slight and is eliminated after the main (i.e., second **chase**) verb has been input. This is apparently because, for these two sentences (and for the grammar), number information does not have any relevance for this task once the main verb has been received.

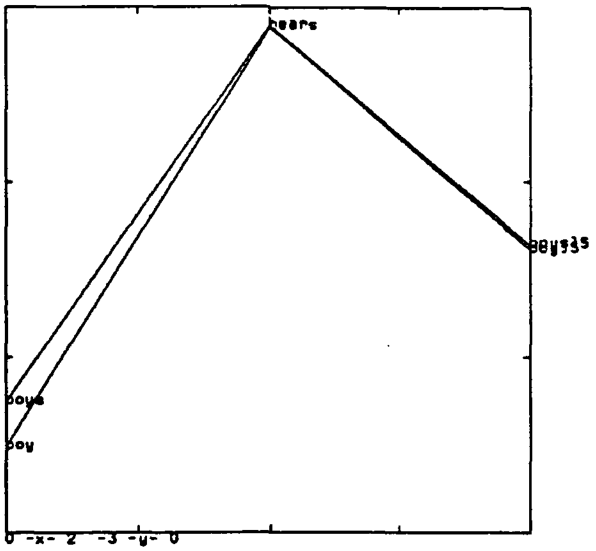


Figure 7
Trajectories through state space for sentences (8a) and (8b). Each point marks the position along the second principle component of hidden unit space, after the indicated word has been input. Magnitude of the second principle component is measured along the ordinate; time (i.e., order of word in sentence) is measured along the abscissa. In this and subsequent graphs the sentence-final word is marked with a]S.

It is not difficult to imagine sentences in which number information may have to be retained over an intervening constituent; sentences (8c) and (8d) are such examples. In both these sentences there is an identical relative clause which follows the initial noun (which differs with regard to number in the two sentences). This material, **who boys chase**, is irrelevant as far as the agreement requirements for the main clause verb. The trajectories through state space for these two sentences have been overlaid and are shown in Figure 8; as can be seen, the differences in the two trajectories are maintained until the main clause

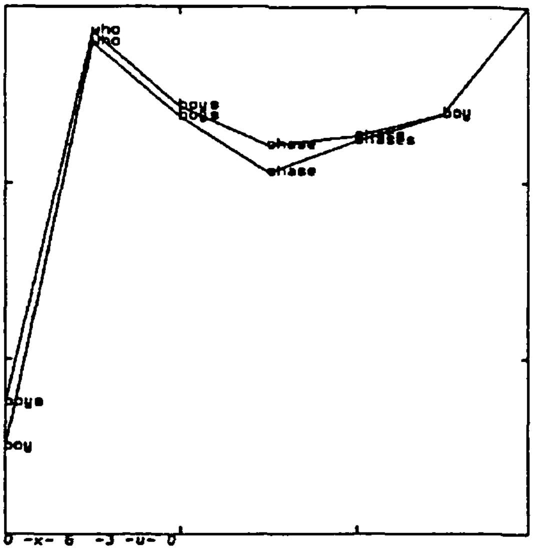


Figure 8
Trajectories through state space for sentences (8c) and (8d).

verb is reached, at which point the states converge.

Verb argument structure

The representation of verb argument structure was examined by probing with sentences containing instances of the three different classes of verbs. Sample sentences are shown in (9).

(9a) boy walks .

(9b) boy sees boy .

(9c) boy chases boy .

The first of these contains a verb which may not take a direct object; the second takes an option direct object; and the third requires a direct object. The movement through state space as these three sentences are processed are shown in Figure 9.

This figure illustrates how the network encodes several aspects of grammatical structure. Nouns are distinguished by role; subject nouns for all three sentences appear in the upper right portion of the space, and object nouns appear below them. (Principal component 4, not shown here, encodes the distinction between verbs and nouns, collapsing across case.) Verbs are differentiated with regard to their argument structure. **Chases** requires a direct object, **sees** takes an optional direct object, and **walks** precludes an object. The difference is reflected in a systematic displacement in the plane of principal components 1 and 3.

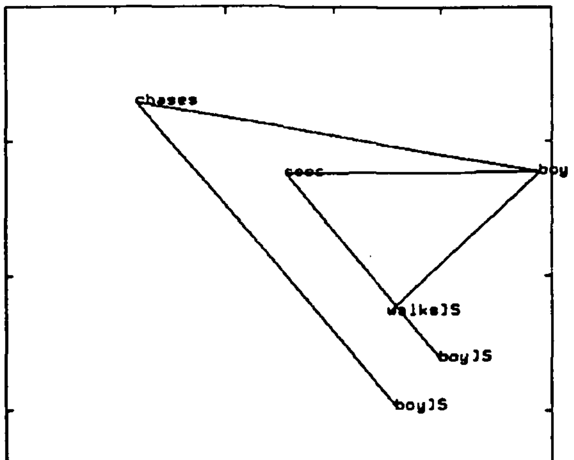


Figure 9

Trajectories through state space for sentences (9a), (9b), and (9c). Principal component 1 is plotted along the abscissa; principal component 3 is plotted along the ordinate.

Relative clauses

The presence of relative clauses introduces a complication into the grammar, in that the representations of number and verb argument structure must be clause-specific. It would be useful for the network to have some way to represent the constituent structure of sentences.

The trained network was given the following sentences.

(10a) boy chases boy .

(10b) boy chases boy who chases boy .

(10c) boy who chases boy chases boy .

(10d) boy chases boy who chases boy who
chases boy .

The first sentence is simple; the other three are instances of embedded sentences. Sentence10a was contained in the training data; sentences 10c, 10d, and 10e were novel and had not been presented to the network during the learning phase.

The trajectories through state space for these four sentences (principal components 1 and 11) are shown in Figure 10. Panel (10a) shows the basic pattern associated with what is in fact the matrix sentences for all four sentences. Comparison of this figure with panels (10b) and (10c) shows that the trajectory for the matrix sentence appears to follow the same for; the matrix subject noun is in the lower left region of state space, the matrix verb appears above

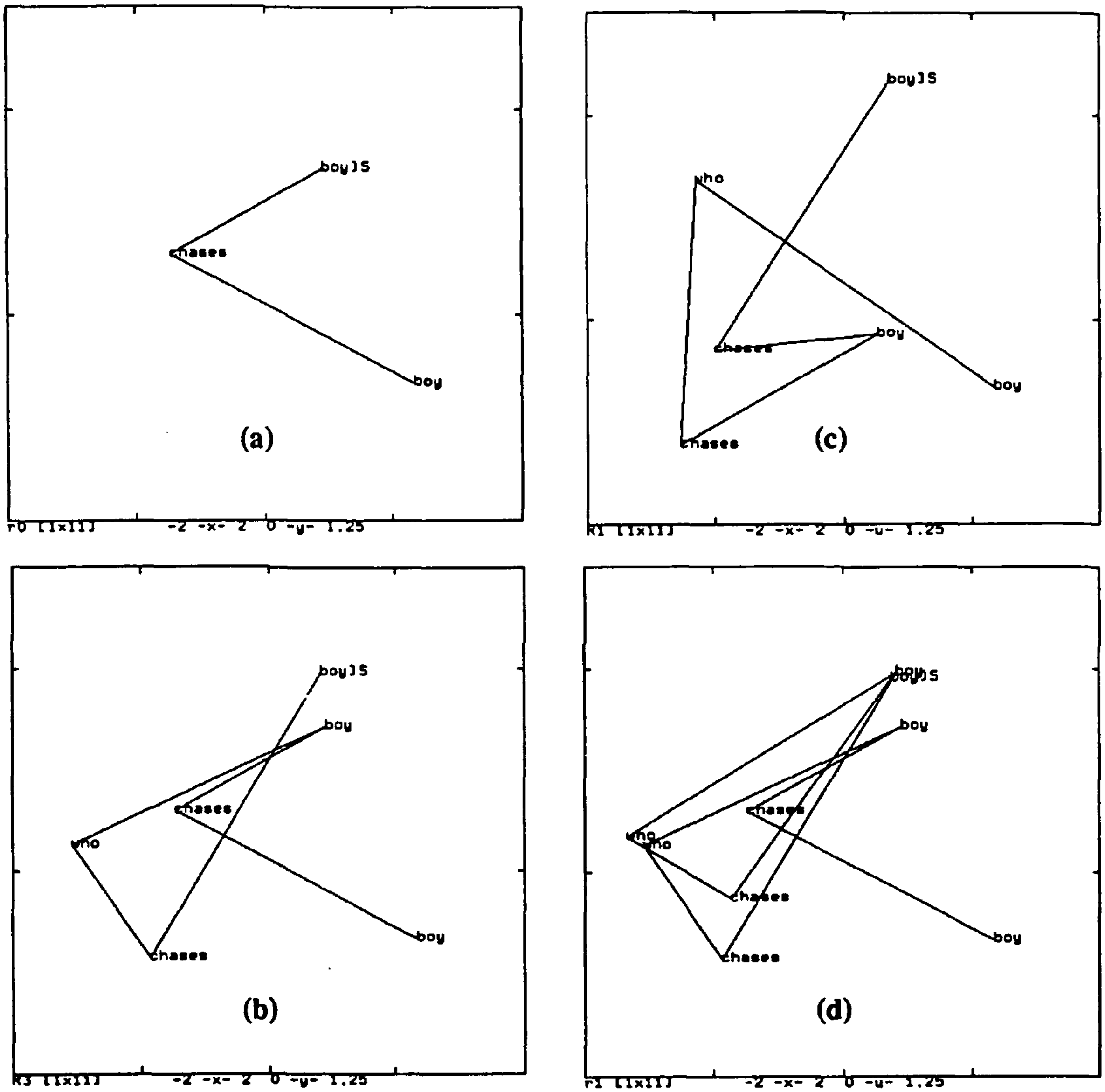


Figure 10

Movement through state space for sentences (10a-d). Principal component 1 is displayed along the abscissa; principal component 11 is displayed along the ordinate it and to the left, and the matrix object noun is near the upper middle region. (Recall that we are looking at only 2 of the 70 dimensions; along other dimensions the noun/verb distinction is preserved categorically.) The relative clause appears involve a replication of this basic pattern, but displaced toward the left and moved slightly downward, relative to the matrix constituents. Moreover, the exact position of the relative clause elements indicates which of the matrix nouns are modified Thus, the relative clause modifying the subject noun is closer to it, and the relative clause modifying the object noun are closer to it. This trajectory pattern was found for all sentences with the same grammatical form; the pattern is thus systematic.

Figure (10d) shows what happens when there are multiple levels of embedding. Successive embeddings are represented in a manner which is similar to the way that the first embedded clause is distinguished from the main clause; the basic patter for the clause is replicated in region of state space which is displaced from the matrix material. This displacement provides a systematic way for the network to encode the depth of embedding in the current state. However, the reliability of the encoding is limited by the precision with which states are represented, which in turn depends on factors such as the number of hidden units and the precision of the numerical values. In the current simulation, the representation degraded after about three levels of embedding. The consequences of this degradation on performance (in the prediction task) are different for different types of sentences. Sentences involving center embedding (e.g., 8c and 8d), in which the level of embedding is crucial for maintaining correct agreement, are more adversely affected than sentences involving so-called tail-recursion (e.g., 10d). In these latter sentences the syntactic structures in principle involve recursion, but in practice the level of embedding is not relevant for the task (i.e., does not affect agreement or verb argument structure in any way).

Figure 10d is interesting in another respect. Given the nature of the prediction task, it is actually not necessary for the network to carry forward any information from prior clauses. It would be sufficient for the network to represent each successive relative clause as an iteration of the previous pattern. Yet the two relative clauses are differentiated. Similarly, Servan-Schreiber, Cleeremans, & McClelland (1988) found that when a simple recurrent network was taught to predict inputs that had been generated by a finite state automaton, the network developed internal representations which corresponded to the FSA states; however, it also redundantly made finer-grained distinctions which encoded the path by which the state had been achieved, even though this information was not used for the task. It thus seems to be a property of these networks that while they are able to encode state in a way which minimizes context as far as behavior is concerned, their nonlinear nature allows them to remain sensitive to context at the level of internal representation.

Part II: Discussion

The basic question addressed in this paper is whether or not connectionist models are capable of complex representations which possess internal structure and which are productively estensible. This question is of particularly of interest with regards to a more general issue: How useful is the connectionist paradigm as a framework for cognitive models? In this context, the nature of representations interacts with a number of other closely related issues. So in order to understand the significance of the present results, it may be useful first to consider briefly two of these other issues. The first is the status of *rules* (whether they exist, whether they are explicit or implicit); the second is the notion of *computational power* (whether it is sufficient, whether it is appropriate).

It is sometimes suggested that connectionist models differ from Classical models in that the latter rely on rules whereas connectionist models are typically not rule systems. Although at first glance this appears to be a reasonable distinction, it is not actually clear that the distinction gets us very far.

The basic problem is that it is not obvious what is meant by a rule. In the most general sense, a rule is a mapping which takes an input and yields an output. Clearly, since many (although not all) neural networks function as input/output systems in which the bulk of the machinery implements some transformation, it is difficult to see how they could not be thought of as rule-systems.

But perhaps what is meant is that the *form* of the rules differs in Classical models and connectionist networks? One suggestion has been that rules are stated *explicitly* in the former, whereas they are only *implicit* in networks. This is a slippery issue, and there is an unfortunate ambiguity in what is meant by implicit or explicit.

One sense of explicit is that a rule is physically present in the system *in its form as a rule*; and furthermore, that that physical presence is important to the correct functioning of the system. However, Kirsh (1989) points out that our intuitions as to what counts as physical presence are highly unreliable and sometimes contradictory. What seems to really be at stake is the speed with which information can be made available. If this is true, and Kirsh argues the point persuasively, then the quality of explicitness does not belong to data structures alone. One must also take into account the nature of the processing system involved, since information in the same form may be easily accessible in one processing system and inaccessible in another.

Unfortunately, our understanding of the information processing capacity of neural networks is quite preliminary. There is a strong tendency in analyzing such networks to view them through traditional lenses. We suppose that if information is not contained in the same form as more familiar computational systems, that information is somehow buried, inaccessible, and implicit. For instance, a network may successfully learn some complicated mapping — say, from text to pronunciation (Sejnowski & Rosenberg, 1987 — but on inspecting the resulting network, it is not immediately obvious how to explain how the mapping works or even to characterize what the mapping is in any precise way. In such cases, it is tempting to say that the network has learned an implicit set of rules. But what we really mean is just that the mapping is "complicated", "difficult to formulate", or "unknown". In fact, this may be a description of our own failure to understand the mechanism rather than a description of the mechanism itself. What is needed are new techniques for network analysis, such as the principal component analysis used in the present work, contribution analysis (Sanger, 1989), weight matrix decomposition (McMillan & Smolensky, 1988), or skeletonization (Mozer & Smolensky, 1989).

If successful, these analyses of connectionist networks may provide us with a new vocabulary for understanding information processing. We may learn new ways in which information can be explicit or implicit, and we may learn new notations for expressing the rules that underlie cognition. The notation of these new connectionist rules may look very different than that used in, for example, production rules. And we may expect that the notation will not lend itself to describing all types of regularity with equal facility.

Thus, the potential important difference between connectionist models and Classical models will not be in whether one or the other systems contains rules, or whether one system encodes information explicitly and the other encodes it implicitly; the difference will lie in the nature of the rules, and in what kinds of information count as explicitly present.

This potential difference brings us to the second issue: computational power. The issue divides into two considerations. Do connectionist models provide *sufficient* computational power (to account for cognitive phenomena); and do they provide the *appropriate* sort of computational power?

The first question can be answered affirmatively with an important qualification. It can be shown that multilayer feedforward networks with as few as one hidden layer, with no squashing at the output and an arbitrary nonlinear activation function at the hidden layer, are capable of arbitrarily accurate approximation of arbitrary mappings. They thus belong to a class of universal approximators (Hornik, Stinchcombe, & White, in press; Stinchcombe & White, 1989). Put simplistically, they are effectively Turing machines. In principle, then, such networks are capable of implementing any function that the Classical system can implement.

The important qualification to the above result is that sufficiently many hidden units be provided. What is not currently known is effect of limited resources on computational power. Since human cognition is carried out in a system with relatively fixed and limited resources, this question is of paramount interest. These limitations provide critical constraints on the nature of the functions which can be mapped; it is an important empirical question whether these constraints explain the specific form of human cognition.

It is in this context that the question of the appropriateness of the computational power becomes interesting. Given limited resources, it is relevant to ask whether the kinds of operations and representations which are naturally made available are those which are likely to figure in human cognition. If one has a theory of cognition which requires sorting of randomly ordered information, e.g., word frequency lists in Forster's (1979) model of lexical access, then it becomes extremely important that the computational framework provide efficient support for the sort operation. On the other hand, if one believes that information is stored associatively, then the ability of the system to do a fast sort is irrelevant. Instead, it is important that the model provide for associative storage and retrieval[1]. Of course, things work in both directions. The availability of certain types of operations may encourage one to build models of a type which are impractical in other frameworks. And the need to work with an inappropriate computational mechanism may blind us from seeing things as they really are.

* * * *

Let us return now to the current work. I would like to discuss first some of the ways in which the work is preliminary and limited. Then I will discuss what I see as the positive contributions of the work. Finally, I would like to relate this work to other connectionist research and to the general question raised at the outset of this discussion: How viable are connectionist models for understanding cognition?

[1] This example was suggested to me by Don Norman.

The results are preliminary in a number of ways. First, one can imagine a number of additional tests that could be performed to test the representational capacity of the simple recurrent network. The memory capacity remains unprobed (but see Servan-Schreiber, Cleeremans, & McClelland, 1988). Generalization has been tested in a limited way (many of the tests involved novels sentences), but one would like to know whether the network can inferentially extend what it knows about the types of noun phrases encountered in the second simulation (simple nouns and relative clauses) to noun phrases with different structures.

Second, while it is true that the agreement and verb argument structure facts contained in the present grammar are important and challenging., we have barely scratched the surface in terms of the richness of linguistic phenomena which characterize natural languages.

Third, natural languages not only contain far more complexity with regard to their syntactic structure, they also have a semantic aspect. Indeed, Langacker (1987) and others have argued persuasively that it is not fruitful to consider syntax and semantics as autonomous aspects of language. Rather, the form and meaning of language are closely entwined. Although there may be things which can be learned by studying artificial languages such as the present one which are purely syntactic, *natural* language processing is crucially an attempt to retrieve meaning from linguistic form. The present work does not address this issue at all, but there are other PDP models which have made progress on this problem (e.g., St. John & McClelland, in press).

What the current work does contribute is some notion of the representational capacity of connectionist models. Various writers (e.g., Fodor & Pylyshyn, 1988) have expressed concern regarding the ability of connectionist representations to encode compositional structure and to provide for open-ended generative capacity. The networks used in the simulations reported here have two important properties which are relevant to these concerns.

First, the networks make possible the development of internal representations which are *distributed* (Hinton, 1988; Hinton, McClelland, Rumelhart, 1986). While not unbounded, distributed representations are less rigidly coupled with resources than localist representations, in which there is a strict mapping between concept and individual nodes.. There is also greater flexibility in determining the dimensions of importance for the model.

Second, the networks studied here build in a sensitivty to context. The important result of the current work is to suggest that the sensitivity to context which is characteristic of many connectionist models, and which is built-in to the architecture of the networks used here, does not preclude the ability to capture generalizations which are at a high level of abstraction. Nor is this a paradox. Sensitivity to context is precisely the mechanism which underlies the ability to abstract and generalize. The fact that the networks here exhibited behavior which was highly regular was not because they learned to be context-insensitive. Rather, they learned to respond to contexts which are more abstractly defined. Recall that even when these networks' behavior seems to ignore context (e.g., Figure 10d; and Servan-Schreiber, Cleeremans, & McClelland, 1988), the internal representations reveal that contextual information is still retained.

This behavior is in striking contrast to that of most Classical models. Representations in Classical models *are* naturally context-insensitive. This insensitivity makes it possible to express generalizations which are fully regular at the highest possible level of representation (e.g., purely syntactic), but they require additional apparatus to account for regularities which reflect the interaction of meaning with form and which are more contextually defined. Connectionist models on the other hand begin the task of abstraction at the other end of the continuum. They emphasize the importance of context and the interaction of form with meaning. As the current work demonstrates, these characteristics lead quite naturally to generalizations at high level of abstraction where appropriate, but the behavior remains ever-rooted in representations which are contextually grounded. The simulations reported here do not capitalize on subtle distinctions in context, but there are ample demonstrations of models which do (e.g., Kawamoto, 1988; McClelland & Kawamoto, 1986; Miikkulainen & Dyer, 1989; St. John & McClelland, in press).

Finally, I wish to point out that the current approach suggests a novel way of thinking about how mental representations are constructed from language input.

Conventional wisdom holds that as words are heard, listeners retrieve lexical representations. Although these representations may indicate the contexts in which the words acceptably occur, the representations are themselves context-free. They exist in some canonical form which is constant across all occurrences. These lexical forms are then used to assist in constructing a complex representation into which the forms are inserted. One can imagine that when complete, the result is an elaborate structure in which not only are the words visible, but which also depicts the abstract grammatical structure which binds those words.

In this account, the process of building mental structures is not unlike the process of building any other physical structure, such as bridges or houses. Words (and whatever other representational elements are involved) play the role of building blocks. As is true of bridges and houses, the building blocks are themselves unaffected by the process of construction.

A different image is suggested in the approach taken here. As words are processed there is no separate stage of lexical retrieval. There are no representations of words in isolation. The representations of words (the internal states following input of a word) always reflect the input taken together with the prior state. In this scenario, words are not building blocks as much as they are cues which guide the network through different grammatical states. Words are distinct from each other by virtue of having different causal properties.

A metaphor which captures some of the characteristics of this approach is the combination lock. In this metaphor, the role of words is analogous to the role played by the numbers in the combination. The numbers have causal properties; they advance the lock into different states. The effect of a number is dependent on its context. Entered in the correct sequence, the numbers move the lock into an open state. The open state may be said to be *functionally compositional* (van Gelder, in press) in the sense that it reflects a particular sequence of events. The numbers are "present" insofar as they are responsible for the final state, but not because they are still physically present.

The limitation of the combination lock is of course that there is only one correct combination. The networks studied here are more complex. The causal properties of the words are highly structure-dependent and the networks allow many "open" (i.e., grammatical) states.

This view of language comprehension emphasizes the functional importance of representations and is similar in spirit to the approach described in Bates & MacWhinney, 1982; McClelland, St. John, & Taraban, 1989; and many others who have stressed the functional nature of language. Representations of language are constructed in order to accomplish some behavior (where, obviously, that behavior may range from day-dreaming to verbal duels, and from to asking directions to composing poetry). The representations are not propositional, and their information content changes constantly over time in accord with the demands of the current task. Words serve as guideposts which help establish mental states that support this behavior; representations are snapshots of those mental states.

ACKNOWLEDGMENTS

I am grateful for many useful discussions on this topic with Jay McClelland, Dave Rumelhart, Elizabeth Bates, Steve Stich, and members of the UCSD PDP/NLP Research Group. I thank McClelland, Mike Jordan, Mary Hare, Ken Baldwin, Gerry Altmann, Colin Brown, and Dennis Norris for critical comments on earlier versions of this paper. This research was supported by contracts N00014-85-K-0076 from the Office of Naval Research, and contract DAAB-07-87-C-H027 from Army Avionics, Ft. Monmouth. Requests for reprints should be sent to the Center for Research in Language, C-008; University of California, San Diego; La Jolla, CA 92093-0108. The author can be reached via electronic mail as elman@amos.ucsd.edu.

REFERENCES

Bates, E., & MacWhinney, B. (1982). Functionalist approaches to grammar. In E. Wanner, & L. Gleitman (Eds.), *Language acquisition: The state of the art.* New York: Cambridge University Press.

Chafe, W. (1970). *Meaning and the Structure of Language*. Chicago: University of Chicago Press.

Dolan, C., & Dyer, M.G. (1987). Symbolic schemata in connectionist memories: Role binding and the evolution of structure. Technical Report UCLA-AI-87-11. Artificial Intelligence Laboratory, University of California, Los Angeles.

Dolan, C.P., & Smolensky, P. (1988). Implementing a connectionist production system using tensor products. Technical Report UCLA-AI-88-15, Artificial Intelligence Laboratory, University of California, Los Angeles.

Elman, J.L. (in press). Finding structure in time. *Cognitive Science.*

Fauconnier, G. (1985). *Mental Spaces.* Cambridge, MA: MIT Press.

Feldman, J. A. & Ballard, D. H., 1982. Connectionist models and their properties. *Cognitive Science, 6,* 205-254.

Fillmore, C.J. (1982). Frame semantics. In *Linguistics in the Morning Calm.* Seoul: Hansin.

Fodor, J. (1976). *The language of thought.* Harvester Press, Sussex.

Fodor, J., & Pylyshyn, Z. (1988). Connectionism and cognitive architecture: A critical analysis. In S. Pinker & J. Mehler (Eds.), *Connections and Symbols.* Cambridge, MA: MIT Press.

Forster, K.I. (1979). Levels of processing and the structure of the language processor. In W.E. Cooper & E. Walker (Eds.) *Sentence processing: Psycholinguistic studies presented to Merrill Garrett.* Hillsdale, N.J.: Lawrence Erlbaum Associates.

Givon, T. (1984). *Syntax: A Functional-Typological Introduction. Volume 1.* Amsterdam: John Benjamins.

Grosjean, F. Spoken word recognition processes and the gating paradigm. *Perception & Psychophysics, 28,* 267-283.

Hanson, S.J., & Burr, D.J. (1987). Knowledge representation in connectionist networks. Bell Communications Research, Morristown, New Jersey.

Hare, M., Corina, D., & Cottrell, G. (1988) Connectionist perspective on prosodic structure. CRL Newsletter, Vol. 3, No. 2. Center for Research in Language, University of California, San Diego.

Hinton, G.E. (1988). Representing part-whole hierarchies in connectionist networks. Technical Report CRG-TR-88-2, Connectionist Research Group, University of Toronto.

Hinton, G.E., McClelland, J.L., & Rumelhart, D.E. (1986). Distributed representations. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition (Vol. 1)* Cambridge, MA: MIT Press.

Hopper, P.J., & Thompson, S.A. (1980). Transitivity in grammar and discourse. *Language, 56,* 251-299.

Hornik, K., Stinchcombe, M., & White, H. (in press). Multi-layer feedforward networks are universal approximators. *Neural Networks.*

Jordan, M. I. (1986). Serial order: A parallel distributed processing approach. Institute for Cognitive Science Report 8604. University of California, San Diego.

Kawamoto, A.H. (1988). Distributed representations of ambiguous words and their resolution in a connectionist network. In S.L. Small, G.W. Cottrell, & M.K. Tanenhaus (Eds.), *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology, and artificial intelligence.* San Mateo, CA: Morgan Kaufmann Publishers.

Kuno, S. (1987). *Functional syntax: Anaphora, discourse and empathy.* Chicago: The University of Chicago Press.

Lakoff, G. (1987). *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind.* Chicago: University of Chicago Press.

Langacker, R.W. (1987). *Foundations of Cognitive Grammar: Theoretical Perspectives. Volume 1.* Stanford: Stanford University Press.

Langacker, R.W. (1988). A usage-based model. *Current Issues in Linguistic Theory, 50,* 127-161.

Marslen-Wilson, W., & Tyler, L.K. (1980). The temporal structure of spoken language understanding. *Cognition, 8,* 1-71

McClelland, J.L. (1987). The case for interactionism in language processing. In M. Coltheart (Ed.), *Attention and performance XII: The psychology of reading.* London: Erlbaum.

McClelland, J.L., St. John, M., & Taraban, R. (1989). Sentence comprehension: A parallel distributed processing approach. Manuscript. Department of Psychology, Carnegie Mellon University..

McMillan, C., & Smolensky, P. (1988). Analyzing a connectionist model as a system of soft rules. Technical Report CU-CS-303-88, Department of Computer Science, University of Colorado, Boulder.

Mozer, M. (1988). A focused back-propagation algorithm for temporal pattern recognition. Technical Report CRG-TR-88-3, Departments of Psychology and Computer Science, University of Toronto.

Mozer, M.C., & Smolensky, P. (1989). Skeletonization: A technique for trimming the fat from a network via relevance assessment. Technical Report CU-CS-421-89, Department of Computer Science, University of Colorado, Boulder.

Oden, G. (1978). Semantic constraints and judged preference for interpretations of ambiguous sentences. *Memory and Cognition, 6,* 26-37.

Pollack, J.B. (1988). Recursive auto-associative memory: Decising compositional distributed representations. *Proceedings of the Tenth Annual Conference of the Cognitive Science Society.* Hillsdale, N.J.: Lawrence Erlbaum

Ramsey, W. (1989). *The philosophical implications of connectionism.* Ph.D. thesis, University of California, San Diego.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning internal representations by error propagation. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition (Vol. 1).* Cambridge, MA: MIT Press.

Salasoo, A., & Pisoni, D.B. (1985). Interaction of knowledge sources in spoken word identification. *Journal of Memory and Language, 24,* 210-231.

Sanger, D. (1989). Contribution analysis: A technique for assigning responsibilities to hidden units in connectionist networks. Technical Report CU-CS-435-89, Department of Computer Science, University of Cclorado, Boulder.

Sejnowski, T.J., & Rosenberg, C.R. (1987). Parallel networks that learn to pronounce English text. *Complex Systems, 1,* 145-168.

Servan-Schreiber, D., Cleeremans, A., & McClelland, J.L. (1988). Encoding sequential structure in simple recurrent networks. CMU Technical Report CMU-CS-88-183. Computer Science Department, Carnegie-Mellon University.

Shastri, L., & Ajjanagadde, V. (1989). A connectionist system for rule based reasoning with multi-place predicates and variables. Technical Report MS-CIS-8905, Computer and Information Science Department, University of Pennsylvania.

Smolensky, P. (1987a). On variable binding and the representation of symbolic structures in connectionist systems. Technical Report CU-CS-355-87, Department of Computer Science, University of Colorado, Boulder.

Smolensky, P. (1987b). On the proper treatment of connectionism. Technical Report CU-CS-377-87, Department of Computer Science University of Colorado, Boulder.

Smolensky, P. (1987c). Putting together connectionism - again. Technical Report CU-CS-378-87, Department of Computer Science, University of Colorado, Boulder.

Smolensky, P. (1988). On the proper treatment of connectionism. *The Behavioral and Brain Sciences, 11.*

St. John, M., & McClelland, J.L. (in press). Learning and applying contextual constraints in sentence comprehension. Technical Report. Department of Psychology. Carnegie-Mellon University. .edn

Stinchcombe, M., & White, H. (1989). Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions. *Proceedings of the International Joint Conference on Neural Networks,* Washington, D.C.

Touretzky, D.S. (1986). BoltzCONS: Reconciling connectionism with the recursive nature of stacks and trees. *Proceedings of the Eight Annual Conference of the Cognitive Science Society.* Hillsdale, N.J.: Lawrence Erlbaum.

Touretzky, D.S. (1989). Rules and maps in connectionist symbol processing. Technical Report CMU-CS-89-158, Department of Computer Science, Carnegie-Mellon University.

Touretzky, D.S., & Hinton, G.E. (1985). Symbols among the neurons: Details of a connectionist inference architecture. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Los Angeles.*

Van Gelder, T.J. (in press). Compositionality: Variations on a classical theme. *Cognitive Science.*